Learning-based Precoding for Zak-OTFS

Naveed Bin Nazir and Ananthanarayanan Chockalingam

***Abstract*—A recently proposed twisted convolution (TC) precoder for Zak-OTFS enables 1-tap equalization at the receiver by maximizing the per-carrier signal-to-interference-plus-noise ratio (SINR). This precoder has noticeable leakage around the target tap which limits performance. In this letter, we tackle this problem in two stages. First, at the transmitter, we propose two learning-based TC precoders, namely, a gradient-based perturbation learning (GBPL) precoder and a liquid neural network (LNN)-based precoder, which suppress the leakage better than the SINR-max precoder, with the LNN precoder producing the strongest localization. Second, at the receiver, unlike conventional receivers that separate channel estimation and symbol detection tasks, we design a lightweight convolutional neural network (CNN) that performs both tasks jointly by simultaneously denoising the received frame and mitigating the residual leakage. Simulation results for fractional delay-Doppler channels show that the proposed precoders outperform the SINR-max precoder, and the CNN receiver offers additional performance gains by efficiently handling remaining leakage.**

***Index Terms*—Zak-OTFS modulation, delay-Doppler domain, twisted convolution precoding, gradient-based learning, liquid neural networks, 1-tap equalization, CNN-based receiver.**

I. Introduction

Zak transform-based orthogonal time frequency space (Zak-OTFS) modulation [1], [2] provides a direct mapping between the delay-Doppler (DD) and time domains, offering strong robustness in doubly-selective channels. Despite its robustness, the Zak-OTFS effective channel typically spreads over many DD taps, requiring joint equalization, often via complex linear/non-linear detection methods. To reduce this burden, the recent work in [3] investigated transmitter-side precoding for Zak-OTFS, for the first time in Zak-OTFS literature. The work in [3] proposes a twisted convolution (TC) filtering-based precoder, which offers two key benefits. First, it shapes the precoded effective channel in a way that enables symbol-by-symbol 1-tap equalization at the receiver relieving the burden of complex joint equalization over all symbols in the frame. Second, the unprecoded effective channel in Zak-OTFS is the cascaded TC of the transmit pulse shaping filter, the physical channel, and the receive pulse shaping filter, and the TC precoding filter naturally fits as another block in the cascade, permitting a TC precoder in closed-form that maximizes per DD bin SINR.

Despite its advantages, the SINR-maximizing (SINR-max) TC precoder in [3] still has noticeable leakage around the target tap (see Fig. 1b) which limits performance. In this letter, we address this limitation by proposing two learning-based precoders under the TC precoding framework along with a lightweight convolutional neural network (CNN) receiver. At the transmitter, the proposed precoders operate in an online manner, where the precoder is obtained for each channel realization via iterative optimization based on the available CSIT, without requiring offline training. At the receiver, the CNN is trained offline and deployed for inference. No joint training between the transmitter and receiver is performed. The main contributions are summarized as follows.

- We first propose a gradient-based perturbation learning (GBPL) precoder, a simple first-order optimization approach that starts from a unit-norm precoder and learns a small complex perturbation that enhances channel localization. The perturbation is updated using the gradient of an energy-focusing loss over several iterations. Despite its simplicity, GBPL yields a more compact precoded effective channel than the SINR-max precoder, which improves symbol error rate (SER) performance.
- To further enhance localization, we introduce a liquid neural network (LNN) [4] based precoder. Instead of learning the perturbation, the LNN learns how to update it at every step, retaining information across iterations through its dynamic state. This enables more accurate updates and yields a more tightly localized precoded effective channel than GBPL, outperforming both SINR-max and GBPL designs.
- Although the proposed precoders significantly reduce the leakage, some residual energy still persists around the target tap. To address this, at the receiver, we design a lightweight CNN that jointly performs channel estimation and symbol detection, denoising the received frame and suppressing the residual leakage with low complexity, yielding further performance gains.

II. TC precoded Zak-OTFS system model

In Zak-OTFS, $\tau_{\rm p}, \nu_{\rm p}$ are the delay-Doppler (DD) periods, with $\tau_{\rm p}\,\nu_{\rm p} = 1$. Let $M \triangleq B\tau_{\rm p}$ and $N \triangleq T\nu_{\rm p}$ be integers, so a frame spans duration $T = N\tau_{\rm p}$ and bandwidth $B = M\nu_{\rm p}$, carrying MN modulation symbols $s[k,l] \in \mathbb{A}$, where $\mathbb{A}$ is the modulation alphabet, for $k = 0, \ldots, M-1$ and $l = 0, \ldots, N-1$. These symbols are extended quasi-periodically across the DD lattice as $s_{\rm dd}[k+nM,\, l+mN] = s[k,l]\, e^{j2\pi n \frac{l}{N}}, n, m \in \mathbb{Z}$. A transmit precoding filter $a[k,l]$ is then applied through discrete twisted convolution to produce the precoded DD-domain signal $x_{\rm dd}[k,l]$, given by $x_{\rm dd}[k,l] = a[k,l] \ast_{\sigma {\rm d}} s_{\rm dd}[k,l]$, where $\ast_{\sigma {\rm d}}$ denotes the discrete twisted convolution[1], which preserves quasi-periodicity. The precoded

Manuscript received May 4, 2026; revised June 16, 2026; accepted July 20, 2026. This work was supported in part by the J. C. Bose National Fellowship and the NSF-DST project (International Cooperation Division), Department of Science and Technology, Government of India. The associate editor coordinating the review of this letter and approving it for publication was Prof. Liqun Fu. (Corresponding author: A. Chockalingam.)

The authors are with the Department of ECE, Indian Institute of Science, Bangalore 560012, India (e-mail: naveedbin@iisc.ac.in, achockal@iisc.ac.in).

[1] $(u \ast_{\sigma {\rm d}} v)[k,l] = \sum_{k',l' \in \mathbb{Z}} u[k-k',\, l-l']\, v[k',l']\, e^{j2\pi \frac{k'(l-l')}{MN}}$.

discrete DD signal $x_{\rm dd}[k,l]$ is mapped to the continuous DD domain using a 2D impulse modulation operation as

$$x_{\rm dd}(\tau,\nu)=\sum\nolimits_{k,l\in\mathbb{Z}} x_{\rm dd}[k,l]\,\delta(\tau-k\tau_p/M)\,\delta(\nu-l\nu_p/N),$$

which is quasi-periodic, i.e., $x_{\rm dd}(\tau+n\tau_p,\,\nu+m\nu_p) = e^{j2\pi n\nu\tau_p}x_{\rm dd}(\tau,\nu)$, $n,m\in\mathbb{Z}$. To restrict the transmit signal to the intended frame duration T and bandwidth B, a DD domain transmit pulse shaping filter $w_{\rm tx}(\tau,\nu)$ is applied, yielding

$$x_{\rm dd}^{w_{\rm tx}}(\tau,\nu)=w_{\rm tx}(\tau,\nu)*_\sigma x_{\rm dd}(\tau,\nu), \tag{1}$$

where $*_\sigma$ denotes the continuous twisted convolution[2]. The corresponding time-domain signal is obtained through inverse Zak transform as $s_{\rm td}(t) = \mathcal{Z}_t^{-1}(x_{\rm dd}^{w_{\rm tx}}(\tau,\nu)) = \sqrt{\tau_p}\int_0^{\nu_p} x_{\rm dd}^{w_{\rm tx}}(t,\nu)\,d\nu$.

The signal $s_{\rm td}(t)$ passes through a doubly-dispersive channel with P paths, modeled as $h(\tau,\nu)=\sum_{i=1}^{P} h_i\,\delta(\tau-\tau_i)\,\delta(\nu-\nu_i)$, where h_i, τ_i, and ν_i denote the gain, delay, and Doppler shift of the ith path. The received signal is $r_{\rm td}(t)=\iint h(\tau,\nu)\,s_{\rm td}(t-\tau)\,e^{j2\pi\nu(t-\tau)}d\tau d\nu + n(t)$, where $n(t)$ denotes the AWGN.

At the receiver, Zak transform[3] is applied to the received time-domain signal to obtain its DD-domain representation as $y_{\rm dd}(\tau,\nu)=\mathcal{Z}_t(r_{\rm td}(t))$, and the output is passed through a receive filter matched to the transmit pulse, given by $w_{\rm rx}(\tau,\nu)=w_{\rm tx}^*(-\tau,-\nu)\,e^{j2\pi\tau\nu}$, yielding

$$y_{\rm dd}^{w_{\rm rx}}(\tau,\nu)=h_{\rm eff}(\tau,\nu)*_\sigma x_{\rm dd}(\tau,\nu)+n_{\rm dd}^{w_{\rm rx}}(\tau,\nu), \tag{2}$$

where $h_{\rm eff}(\tau,\nu) = w_{\rm rx}(\tau,\nu)*_\sigma h(\tau,\nu)*_\sigma w_{\rm tx}(\tau,\nu)$ is the effective DD domain channel response (a.k.a. the 'effective channel') and $n_{\rm dd}^{w_{\rm rx}}(\tau,\nu)$ is the filtered DD domain noise. Sampling (2) on the information grid points $\tau=\frac{k\tau_{\rm p}}{M},\nu=\frac{l\nu_{\rm p}}{N}$ yields the discrete model

$$y_{\rm dd}[k,l]=h_{\rm eff}[k,l]*_{\sigma{\rm d}} x_{\rm dd}[k,l]+n_{\rm dd}[k,l]. \tag{3}$$

Substituting the precoded signal into (3), we obtain

$$\begin{aligned} y_{\rm dd}[k,l] &= h_{\rm eff}[k,l]*_{\sigma{\rm d}} a[k,l]*_{\sigma{\rm d}} s_{\rm dd}[k,l]+n_{\rm dd}[k,l]\\ &= h_{\rm a}[k,l]*_{\sigma{\rm d}} s_{\rm dd}[k,l]+n_{\rm dd}[k,l], \end{aligned} \tag{4}$$

where $h_{\rm a}[k,l]\triangleq w_{\rm rx}[k,l]*_{\sigma{\rm d}} h[k,l]*_{\sigma{\rm d}} w_{\rm tx}[k,l]*_{\sigma{\rm d}} a[k,l]$, equivalently written as $h_{\rm a}[k,l]=h_{\rm eff}[k,l]*_{\sigma{\rm d}} a[k,l]$, denotes the effective precoded DD-domain channel response (i.e., the precoded effective channel). Equation (4), therefore, gives the discrete I/O relation of the TC-precoded Zak-OTFS system.

Support sets of $h_{\rm eff}[k,l]$***,*** $h_a[k,l]$***, and*** $a[k,l]$***:*** Let the support of $h_{\rm eff}[k,l]$, denoted by $\mathcal{S}_{h_{\rm eff}}$, be defined as $\mathcal{S}_{h_{\rm eff}}\triangleq\{(k,l)\,|\,k_{\min}\le k\le k_{\max},\ l_{\min}\le l\le l_{\max}\}$, where $k_{\min}\le 0\le k_{\max}$, $(k_{\max}-k_{\min})<M$, and $0\le l_{\max}<N/2$. The precoding filter support $\mathcal{S}_a$ is selected such that the resulting precoded channel $h_a[k,l]$ has support $\mathcal{S}_{h_a}\triangleq\{(k,l)\mid -\frac{M}{2}\le k\le\frac{M}{2}-1,\ -\frac{N}{2}\le l\le\frac{N}{2}-1\}$. Accordingly, $\mathcal{S}_a$ is chosen as

$$\mathcal{S}_a\triangleq\left\{(k,l)\;\middle|\;\begin{matrix}-M/2-k_{\min}\le k\le M/2-k_{\max}-1,\\ -N/2+l_{\max}\le l\le N/2-l_{\max}-1\end{matrix}\right\}. \tag{5}$$

I/O relation in matrix-vector form: The end-to-end DD I/O relation in (4) can be expressed in matrix-vector form as

$$\mathbf{y}=\mathbf{H}_{\rm eff}\mathbf{s}+\mathbf{n}, \tag{6}$$

where $\mathbf{s},\mathbf{y},\mathbf{n}\in\mathbb{C}^{MN\times 1}$ with entries $s_{kN+l+1}=s_{\rm dd}[k,l]$, $y_{kN+l+1}=y_{\rm dd}[k,l]$, and $n_{kN+l+1}=n_{\rm dd}[k,l]$. The matrix $\mathbf{H}_{\rm eff}\in\mathbb{C}^{MN\times MN}$ denotes the precoded effective channel with

$$\mathbf{H}_{\rm eff}[k'N+l'+1,\ kN+l+1]=\begin{cases} h_{\rm a}[k'-k,\ l'-l]\,e^{j2\pi\frac{(l'-l)k}{MN}},\\ \quad (k'-k,\ l'-l)\in\mathcal{S}_{h_{\rm a}}\\ 0,\quad \text{otherwise,}\end{cases}$$

where $k',k=0,\dots,M-1$ and $l',l=0,\dots,N-1$. Assuming i.i.d. zero-mean symbols $s[k,l]$ with energy E, i.e., $\mathbb{E}[\,|h_{\rm a}[k,l]*_{\sigma{\rm d}} s_{\rm dd}[k,l]|^2\,]=E$, unit-energy transmit and matched receive filters, and a normalized precoding filter satisfying $\sum_{(k,l)\in\mathcal{S}_a}|a[k,l]|^2=1$, the received average energy per DD bin becomes E. The DD noise variance is N_0, leading to a per-bin signal-to-noise ratio (SNR) of $\rho\triangleq E/N_0$.

Precoder vector: Arranging the MN taps of $h_{\rm a}[k,l]$ into a vector $\mathbf{h}_{\rm a}$, we write $\mathbf{h}_{\rm a}=\big(h_{\rm a,0}\,,\,h_{\rm a,1}\,,\,\cdots\,,\,h_{\rm a,(MN-1)}\big)^T$, where $h_{{\rm a},k+\frac{M}{2}+(l+\frac{N}{2})M}=h_{\rm a}[k,l]$, $(k,l)\in\mathcal{S}_{h_{\rm a}}$. From the discrete twisted convolution definition, it follows that $\mathbf{h}_{\rm a}=\mathbf{H}'\mathbf{a}$, where $\mathbf{a}\in\mathbb{C}^{K\times 1}$, $K=(k_{\min}-k_{\max}+M)(N-2l_{\max})$, is the precoder vector consisting of the precoding filter taps given by $\mathbf{a}=(a_0\,,\,a_1\,,\,\cdots\,,a_{K-1})^T$, where $a_{k+\frac{M}{2}+k_{\min}+\left(l+\frac{N}{2}-l_{\max}\right)(M-k_{\max}+k_{\min})}\triangleq a[k,l]$ with $(k,l)\in\mathcal{S}_a$. The matrix $\mathbf{H}'\in\mathbb{C}^{MN\times K}$ has entries in its $r(k,l)$th row and $c(k',l')$th column with $(k,l)\in\mathcal{S}_{h_{\rm a}}$, $(k',l')\in\mathcal{S}_a$, $r(k,l)\triangleq k+\frac{M}{2}+(l+\frac{N}{2})M$, $c(k',l')\triangleq k'+\frac{M}{2}+k_{\min}+(l'+\frac{N}{2}-l_{\max})(M-k_{\max}+k_{\min})$, as

$$H'_{r(k,l),\,c(k',l')}\triangleq\begin{cases} h_{\rm eff}[k-k',l-l']\,e^{j2\pi k'\frac{(l-l')}{MN}},\\ \qquad (k-k',l-l')\in\mathcal{S}_{h_{\rm eff}},\\ 0,\quad \text{otherwise.}\end{cases} \tag{7}$$

1-tap channel estimator/equalizer at the receiver: When the system operates well inside the crystalline regime [2], i.e., $(k_{\max}-k_{\min})\ll M$ and $2l_{\max}\ll N$, where $k_{\max}\triangleq\lceil B\tau_{\max}\rceil$, $l_{\max}\triangleq\lceil 2T\nu_{\max}\rceil$, and $k_{\min}=-k_{\max}$, $l_{\min}=-l_{\max}$, the delay and Doppler periods become much larger than their corresponding channel spreads. In this case, the I/O relation becomes predictable, and the entire channel can be inferred from a single pilot measurement. Since the precoding filter concentrates most of the effective channel energy at the origin, the downlink receiver only needs to estimate $h_{\rm a}[0,0]$. Thus, a single pilot symbol without any guard region is sufficient, leading to high spectral efficiency. The pilot is placed at $(k_{\rm p},l_{\rm p})=(M/2,N/2)$, and all remaining bins carry data, with $s[k_{\rm p},l_{\rm p}]=\sqrt{\eta(MN-1)E}$, where η is the pilot-to-data power ratio (PDR). The pilot and data energies are $\eta(MN-1)E$ and $(MN-1)E$, respectively. The estimate of $h_{\rm a}[0,0]$ from the received pilot $y_{\rm dd}[k_{\rm p},l_{\rm p}]$ is then obtained as $\widehat{h}_{\rm a}[0,0]\triangleq y_{{\rm dd}[k_{\rm p},l_{\rm p}]}/\big(\sqrt{\eta(MN-1)E}\,(1+1/(\rho\eta(MN-1)))\big)$. Since the precoding filter already mitigates most of the channel spread, the interference across precoded taps becomes small, making 1-tap equalization adequate. The equalized (k,l)th information symbol is therefore computed as $\widehat{s}[k,l]=\widehat{h}_{\rm a}^*[0,0]\,y_{\rm dd}[k,l]/(|\widehat{h}_{\rm a}[0,0]|^2+1/\rho)$.

III. Learning-Based Precoders and CNN receiver

The TC precoder in [3] is designed by maximizing the per-bin SINR at the downlink receiver. In contrast, we adopt a learning-based formulation in which the precoder is obtained by minimizing an energy-focusing loss. For this, we define a

[2] $(u*_\sigma v)(\tau,\nu)=\iint u(\tau',\nu')\,v(\tau-\tau',\nu-\nu')\,e^{j2\pi\nu'(\tau-\tau')}d\tau' d\nu'$.
[3] $\mathcal{Z}_t(x(t))=\sqrt{\tau_p}\sum_{k\in\mathbb{Z}} x(\tau+k\tau_p)\,e^{-j2\pi\nu k\tau_p}$.

Algorithm 1 Proposed GBPL precoder

Require: $\mathbf{H}'$, $\mathbf{h}_{\text{target}}$, learning rate α, no. of iterations T
Ensure: Learned perturbations $\boldsymbol{\delta}_{\text{Re}}^{(T)}$, $\boldsymbol{\delta}_{\text{Im}}^{(T)}$ (used to form $\mathbf{a}^*$)
Initialize $\mathbf{a}_{\text{init}}$ (unit-norm, randomly generated)
Initialize $\boldsymbol{\delta}_{\text{Re}}^{(0)}$, $\boldsymbol{\delta}_{\text{Im}}^{(0)} \sim \mathcal{N}(0, 0.01)$
for $n = 0$ to $T-1$ **do**
Form perturbed vector: $\tilde{\mathbf{a}}^{(n)} = \mathbf{a}_{\text{init}} + (\boldsymbol{\delta}_{\text{Re}}^{(n)} + j\boldsymbol{\delta}_{\text{Im}}^{(n)})$
Normalize: $\mathbf{a}^{(n)} \leftarrow \tilde{\mathbf{a}}^{(n)}/\|\tilde{\mathbf{a}}^{(n)}\|$
Compute loss: $\mathcal{L}(\mathbf{a}^{(n)})$
Compute gradients w.r.t perturbations: $\nabla_{\boldsymbol{\delta}_{\text{Re}}}\mathcal{L}, \nabla_{\boldsymbol{\delta}_{\text{Im}}}\mathcal{L}$
Update perturbations (Adam):
$\boldsymbol{\delta}_{\text{Re}}^{(n+1)} \leftarrow \boldsymbol{\delta}_{\text{Re}}^{(n)} - \alpha\,\nabla_{\boldsymbol{\delta}_{\text{Re}}}\mathcal{L}, \quad \boldsymbol{\delta}_{\text{Im}}^{(n+1)} \leftarrow \boldsymbol{\delta}_{\text{Im}}^{(n)} - \alpha\,\nabla_{\boldsymbol{\delta}_{\text{Im}}}\mathcal{L}$
end for
return $\boldsymbol{\delta}_{\text{Re}}^{(T)}$, $\boldsymbol{\delta}_{\text{Im}}^{(T)}$ $\left(\mathbf{a}^* = \frac{\mathbf{a}_{\text{init}}+\boldsymbol{\delta}_{\text{Re}}^{(T)}+j\boldsymbol{\delta}_{\text{Im}}^{(T)}}{\left\|\mathbf{a}_{\text{init}}+\boldsymbol{\delta}_{\text{Re}}^{(T)}+j\boldsymbol{\delta}_{\text{Im}}^{(T)}\right\|}\right)$

target response $h_{\text{target}}[k,l]$ that concentrates all the effective channel energy at the origin, i.e.,

$$h_{\text{target}}[k,l] \triangleq \begin{cases} \sum_{(k',l')\in\mathcal{S}_{h_{\text{eff}}}} |h_{\text{eff}}[k',l']|^2, & (k,l)=(0,0) \\ 0, & \text{otherwise.} \end{cases}$$

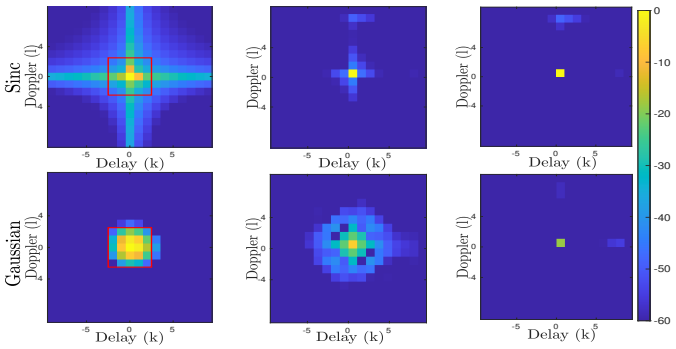
representing the ideal fully localized channel. We then seek a unit-norm precoder $\mathbf{a}$ so that the precoded response $\mathbf{h}_{\text{a}} = \mathbf{H}'\mathbf{a}$ aligns closely with $\mathbf{h}_{\text{target}}$, measured using the following loss:

$$\mathcal{L}(\mathbf{a}) = 1 - |\langle \mathbf{h}_{\text{a}}, \mathbf{h}_{\text{target}}\rangle|^2 / \left(\|\mathbf{h}_{\text{a}}\|^2 \, \|\mathbf{h}_{\text{target}}\|^2 + \varepsilon\right), \quad (8)$$

where $\varepsilon > 0$ prevents division by zero. This loss becomes zero when the precoded channel perfectly matches the desired target localized pattern, and it serves as the optimization objective for the learning-based precoder designs.

Gradient-based perturbation learning (GBPL) precoder: GBPL is a first-order optimization approach that refines an initial unit-norm complex vector to minimize the energy-focusing loss. We initialize a precoder $\mathbf{a}_{\text{init}} \in \mathbb{C}^{K\times 1}$ as a random unit-norm vector and introduce a learnable perturbation $\boldsymbol{\delta} = \boldsymbol{\delta}_{\text{Re}} + j\boldsymbol{\delta}_{\text{Im}}$, where the real and imaginary parts lie in $\mathbb{R}^K$, yielding $2K$ trainable parameters. At each iteration, the perturbed candidate $\tilde{\mathbf{a}} = \mathbf{a}_{\text{init}} + \boldsymbol{\delta}$ is normalized, the loss (8) is evaluated, and the perturbation is updated using its gradient through Adam optimizer (adaptive moment estimation). Repeating this process over multiple iterations concentrates channel energy more effectively than the SINR-max precoder. The GBPL precoder design is listed in **Algorithm 1**.

Liquid neural network-based precoder: While the GBPL precoder improves localization compared to the SINR-max design, its updates rely solely on first-order gradients. To enable more adaptive refinement, we employ an LNN that models the update rule as a continuous-time dynamical system with neuron-specific time constants. Unlike recurrent neural networks (RNNs) / long short-term memory networks (LSTMs) that suffer from vanishing gradients on long sequences, LNNs retain relevant information through gated ordinary differential equations (ODE)-driven state evolution [4]. The hidden state $\mathbf{h}(t)$ evolves as $d\mathbf{h}(t)/dt = -\big(1/\boldsymbol{\tau} + \mathbf{g}(t)\big)\odot\mathbf{h}(t) + \mathbf{g}(t)\odot\mathbf{h}_{\text{ss}}$, where $\boldsymbol{\tau}$ is a learnable time-constant vector, $\mathbf{g}(\cdot)$ is a gating function, $\mathbf{h}_{\text{ss}}$ is the steady-state vector. Using a fused implicit-explicit Euler solver with step size $\Delta t \in \mathbb{R}$, the discrete update becomes $\mathbf{h}(t+\Delta t) = \big(\mathbf{h}(t) + \Delta t\,\mathbf{g}(t)\odot\mathbf{h}_{\text{ss}}\big)/\big(\mathbf{1} + \Delta t\big(1/\boldsymbol{\tau} + \mathbf{g}(t)\big)\big)$, where $\mathbf{g}(t) = \tanh(\mathbf{W}\mathbf{h}(t) + \mathbf{U}\mathbf{x}(t))$, $\mathbf{W}, \mathbf{U} \in \mathbb{R}^{h_{\text{L}}\times h_{\text{L}}}$, $\boldsymbol{\tau}, \mathbf{h}_{\text{ss}} \in \mathbb{R}^{h_{\text{L}}}$ and $\mathbf{x}(t)$ denotes the input feature vector to the LNN; all are randomly initialized and learned during training, and all operations are applied elementwise. These dynamics produce smooth updates across iterations, which is desirable when learning how to update the precoder vector.



(a) Without precoding (b) SINR-max prec. [3] (c) LNN prec. (Prop)

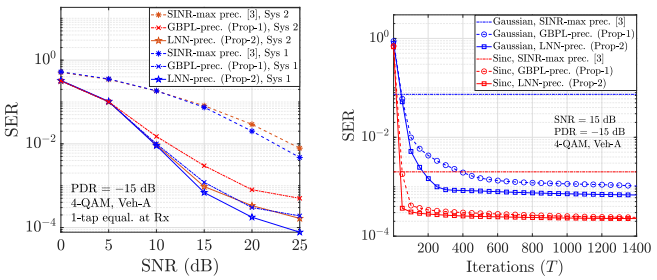
Fig. 1: Effective-channel heatmaps comparing unprecoded, SINR-max, and LNN precoding at $\rho = 15$ dB.

We initialize a randomly generated unit-norm vector $\mathbf{a}^{(0)} \in \mathbb{C}^{K\times 1}$ and compute the gradient of the loss (8). Its real and imaginary parts are concatenated into a $2K$-dimensional input feature. Feeding this to an LNN requires a large hidden dimension, so we adopt a compact FC–LNN–FC structure: the first FC (fully-connected) layer extracts gradient features and reduces dimensionality, the LNN captures temporal dependencies across update steps, and the final FC layer outputs the refined perturbation vector $\Delta\mathbf{p}^{(n)} \in \mathbb{R}^{2K}$. This architecture contains $2h_{\text{L}}^2 + 4Kh_{\text{L}} + 3h_{\text{L}} + 2K$ parameters, where h_{L} is the LNN hidden-state size. The resulting LNN optimizer learns an adaptive update policy that evolves over iterations, achieving stronger channel localization than both SINR-max and GBPL precoders. The LNN precoder design is in **Algorithm 2**.

Performance comparison between SINR-max and proposed precoders: Figure 1 compares the energy localization achieved using the SINR-max precoder and the proposed LNN precoder with sinc and Gaussian pulse shaping filters at $\rho = 15$ dB. These two pulses represent contrasting cases, with the sinc pulse satisfying the Nyquist property and favoring interference-free detection, while the Gaussian pulse is well localized but introduces inter-symbol interference due to non-zero values at the neighboring sampling points. The system parameters are: Vehicular-A (Veh-A) channel model with $P = 6$ paths, maximum delay $\tau_{\max} = 2.51$ μs, maximum Doppler $\nu_{\max} = 1$ kHz, and power delay profile as given in Table 2 in [2]. The Doppler associated with the ith path is modeled as $\nu_i = \nu_{\max}\cos(\theta_i)$, where θ_i are independently and uniformly distributed in $[0, 2\pi)$. With $\nu_p = 30$ kHz and $\tau_p = 1/\nu_p = 33.33$ μs, we set $M = N = 18$, giving a bandwidth of $B = M\nu_p = 540$ kHz and frame duration $T = N\tau_p = 0.6$ ms. While the SINR-max precoder (Fig. 1b) achieves moderate localization with sinc pulse, the achieved concentration is still weak. Also, the Gaussian pulse has substantial leakage, clearly visible from the wider heatmap spread. In contrast, the proposed LNN precoder concentrates almost

Algorithm 2 Proposed LNN precoder

Require: $\mathbf{H}'$, $\mathbf{h}_{\text{target}}$, learning rate α, no. of iterations T
Ensure: Optimized precoder $\mathbf{a}^*$
Initialize precoder $\mathbf{a}^{(0)}$ (unit-norm) and hidden state $\mathbf{h}^{(0)}$
Initialize FC and LNN parameters $\{\mathbf{W}, \mathbf{U}, \boldsymbol{\tau}, \mathbf{h}_{\text{ss}}\}$
for $n = 0$ to $T-1$ **do**
Compute loss gradient: $\boldsymbol{\nabla}^{(n)} = \nabla_{\mathbf{a}} \mathcal{L}(\mathbf{a}^{(n)})$
Form input vector: $\mathbf{u}^{(n)} = [\Re\{\boldsymbol{\nabla}^{(n)}\}; \Im\{\boldsymbol{\nabla}^{(n)}\}] \in \mathbb{R}^{2K}$
Feature extraction (FC): obtain $\mathbf{x}^{(n)}$ from $\mathbf{u}^{(n)}$
LNN update: compute hidden state $\mathbf{h}^{(n+1)}$
Output: obtain perturbation $\Delta\mathbf{p}^{(n)}$ from output FC layer
Conversion: $\Delta\mathbf{a}^{(n)} = \Delta\mathbf{p}^{(n)}_{1:K} + j\,\Delta\mathbf{p}^{(n)}_{K+1:2K}$
Update precoder: $\mathbf{a}^{(n+1)} = \dfrac{\mathbf{a}^{(n)} - \alpha\,\Delta\mathbf{a}^{(n)}}{\|\mathbf{a}^{(n)} - \alpha\,\Delta\mathbf{a}^{(n)}\|}$
end for
return $\mathbf{a}^* = \mathbf{a}^{(T)}$



(a) (b)

Fig. 2: (a) SER vs data SNR for Gaussian pulse. (b) SER vs number of iterations (T) for Gaussian and sinc pulses.

all channel energy at the central tap with only minor leakage for both pulse shapes, demonstrating stronger localization. To quantify this behavior, we define the residual leakage energy $\mathcal{E}_{\text{res}} = \sum_{(k,l)\in\mathcal{F}\setminus\mathcal{C}} |h_a[k,l]|^2 \;/\; \sum_{(k,l)\in\mathcal{F}} |h_a[k,l]|^2$, where $\mathcal{C}$ is the center DD tap and $\mathcal{F}$ is the fundamental period (red box). With SINR-max precoder, the $\mathcal{E}_{\text{res}}$ at $\rho = 15$ dB for sinc and Gaussian pulses are 4.03% and 20.38%, respectively. The corresponding $\mathcal{E}_{\text{res}}$ values for the LNN precoder are 2.61% (sinc) and 3.73% (Gaussian), highlighting the superior localization achieved by the proposed LNN precoder.

Figure 2(a) compares the SER performance of the SINR-max, GBPL, and LNN precoders for Gaussian pulse with 1-tap equalizer-based receiver under two scenarios: System 1, a moderate Doppler spread scenario ($\nu_{\max} = 1$ kHz, $M = N = 18$) and System 2, a higher Doppler spread scenario ($\nu_{\max} = 2$ kHz, $M = N = 24$). It is observed that both GBPL and LNN precoders significantly outperform the SINR-max precoder. Moreover, the LNN precoder performs better than the GBPL precoder, with the performance gap becoming more pronounced under higher Doppler spreads. Figure 2(b) shows the convergence behavior of the proposed precoders for Gaussian and sinc filters with 1-tap receiver. It is observed that $i)$ sinc filter performs better than Gaussian filter (because of better localization – see Fig. 1), $ii)$ LNN precoder converges to a better SER compared to GBPL precoder, and $iii)$ LNN precoder converges faster than GBPL precoder.

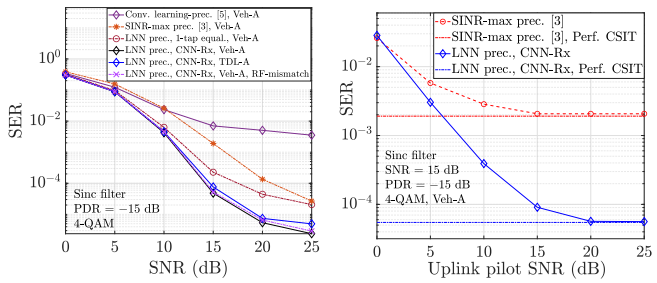
Complexity analysis: For both GBPL and LNN precoders, the dominant operation in evaluating the loss is the matrix–vector product $\mathbf{H}'\mathbf{a}$ at each iteration. Thus, the overall complexity of the GBPL precoder over T iterations is $\mathcal{O}(TMNK)$. The LNN complexity includes an additional $\mathcal{O}(Kh_{\text{L}} + h_{\text{L}}^2)$ from the FC layers and from computing $\mathbf{g}(t) = \tanh(\mathbf{W}\mathbf{h}(t) + \mathbf{U}\mathbf{x}(t))$ leading to a total complexity of $\mathcal{O}(T(MNK + Kh_{\text{L}} + h_{\text{L}}^2))$. In contrast, the closed-form SINR-max precoder $\mathbf{a}_{\text{SINR}} = (\mathbf{I}/\rho + \sum_{i\neq p} \mathbf{h}'_i \mathbf{h}'^H_i)^{-1}\mathbf{h}'_p$, where $p = M(N+1)/2$ and $\mathbf{h}'^H_i$ is the ith row of $\mathbf{H}'$, requires forming $(MN-1)$ rank-1 matrices and accumulating them into a $K \times K$ matrix, followed by matrix inversion, resulting in a computational cost of $\mathcal{O}(MNK^2 + K^3)$, which scales cubically with K. The complexities of the precoders are summarized in Table I. For $M = N = 18$, $K = 196$, $h_{\text{L}} = 64$, the SINR-max precoder has a fixed complexity of 7.001×10^7 real operations (ROs), whereas the per-iteration complexities of the GBPL and LNN precoders are 1.542×10^6 and 3.434×10^6 ROs, respectively. From Fig. 2(b), we see that both precoders converge at about 100 iterations for sinc and 500 iterations for Gaussian. For LNN precoder, this corresponds to total complexities of about 3.434×10^8 ROs for sinc and 17.17×10^8 ROs for Gaussian, i.e., the improved performance of LNN compared to SINR-max (by more than an order of magnitude in SER) comes at the cost of more complexity[4], indicating a performance-complexity tradeoff at play.

TABLE I: Per-iteration complexity in number of ROs.

Precoder	Real additions (per iter.)	Real multiplications (per iter.)
SINR-max	$4K^3/3 + (2MN+2)K^2 - 3K$	$4K^3/3 + (2MN+2)K^2 - 4K$
GBPL	$12MNK + 12K + 18MN$	$12MNK + 12K + 24MN$
LNN	$6h_{\text{L}}^2 + 12Kh_{\text{L}} + 12h_{\text{L}} + 24MNK + 18K + 36MN$	$6h_{\text{L}}^2 + 12Kh_{\text{L}} + 12h_{\text{L}} + 24MNK + 21K + 48MN$

CNN-based receiver: Although the proposed learning-based precoders significantly localize the effective channel, a small amount of leakage still remains. To suppress this residual energy and further enhance equalization/detection performance, we replace the two-stage receiver (1-tap channel estimation followed by symbol detection) with a single lightweight CNN. The network not only enhances channel localization but also denoises the received frame and directly maps it to the transmitted symbols. The training dataset is generated using (6), i.e., the received signal already incorporates the effect of the precoder through the effective channel. The CNN thus learns a direct end-to-end mapping from the received signal to the transmitted data symbols, implicitly accounting for the modified channel structure. Each input sample is $\mathbf{r}^{(i)}_{\text{train}} \in \{\Re(\mathbf{y}^{(i)}), \Im(\mathbf{y}^{(i)})\}$ and the corresponding target label is the original transmitted symbol vector $\mathbf{y}^{(i)}_{\text{train}} \in \{\Re(\mathbf{s}^{(i)}), \Im(\mathbf{s}^{(i)})\}$, treating the real and imaginary components as separate input-output pairs. The CNN is trained offline using simulated channel realizations based on the Veh-A channel model. The

[4]We note that since the precoder learning/optimization is implemented on the base station side, where greater computational resources are available compared to user equipment (UE) side, these complexities can be affordable. Importantly, this aids to reduce the complexity on the resource-constrained UE side to work with a 1-tap equalizer. Also, precoder complexity reduction without compromising much on performance is open for further investigation.



(a) (b)

Fig. 3: (a) SER vs data SNR. (b) SER vs uplink pilot SNR.

proposed CNN structure: *Input (18×18×1) → Conv(3×3, 8) → SiLU → Conv(3×3, 16, stride 2) → SiLU → Bottleneck: Conv(3×3, 16) → SiLU → TransposedConv(3×3, 16, stride 2) → SiLU → Conv(3×3, 8) → SiLU → Conv(1×1, 1) → Output.* This corresponds to a lightweight architecture with only 7057 trainable parameters. The 1-tap equalizer involves simple element-wise operations with $\mathcal{O}(MN)$ complexity per frame. The proposed CNN-based receiver also scales linearly with the DD grid size MN. Specifically, each convolutional layer requires approximately $k^2 C_{\text{in}} C_{\text{out}} MN$ operations, where k, C_{in}, and C_{out} denote the kernel size and the number of input and output channels, respectively. Summing across all layers, the total complexity remains on the order of 10^5 operations per frame for $M = N = 18$, which is moderate for the good denoising/leakage suppression it achieves. It is trained with 320,000 samples at -15 dB PDR and 15 dB SNR and RMSE loss for 100 epochs, with a learning rate of 5×10^{-4}.

IV. Results and Discussion

We consider a TDD system where an uplink pilot frame is transmitted at 20 dB SNR to estimate CSIT at the BS[5]. The system parameters are the same as in Fig. 1. On the downlink, an embedded pilot frame is used with a pilot symbol in the middle bin and data symbols in all the remaining bins (since precoding strongly suppresses pilot–data interference, no guard region is required) and $\eta = -15$ dB is used. For learning, $\alpha = 10^{-3}$, $\Delta t = 0.01$ and $T = 1500$ are used for both GBPL and LNN precoders.

Figure 3(a) shows the SER performance of the SINR-max and the proposed LNN precoder with sinc filter for 1-tap equalization-based receiver and CNN-based receiver. For comparison, we evaluated a conventional learning-based precoding approach that learns a direct channel-to-precoder mapping via offline training, similar to prior data-driven designs [5]. In the Zak-OTFS setting, this approach exhibits poor convergence and saturates at a high error floor of about 3×10^{-3}. In contrast, the proposed learning-based precoding approach achieves much better performance. The LNN precoder outperforms the SINR-max precoder, e.g., at $\rho = 15$ dB, the SER improves by nearly one order of magnitude (about 2×10^{-4} vs 2×10^{-3}). With the proposed CNN-based receiver, the SER further drops to about 5×10^{-5}. At high SNRs, while all precoders exhibit a similar error floor due to residual leakage, the CNN receiver noticeably lowers this floor by jointly suppressing leakage and noise. Also, the CNN receiver is found to generalize well under channel model mismatches, e.g., robust performance is achieved when the CNN trained with Veh-A channel is tested with 3GPP TDL-A channel consisting of $P = 23$ paths [6] and $\nu_{\max} = 1$ kHz. This can be attributed to the effectiveness of the precoding, which strongly localizes the effective channel energy into a dominant DD bin, thereby presenting a similar localized channel structure to the CNN across different channel models. The proposed learning approach is also found to be robust against RF reciprocity mismatches, such as Tx-Rx imbalance [7], e.g., the performance degradation in the presence of Tx-Rx imbalance[6] is marginal. This is because 1) the LNN could absorb the Rx chain imbalance in the online learning of the precoder, and 2) the CNN could alleviate the effect of the Tx chain imbalance. Figure 3(b) presents the SER as a function of uplink pilot SNR, capturing the effect of CSIT errors. It is seen that 1) the SER improves as the uplink pilot SNR increases (as expected) and gets close to the performance with perfect CSIT, and 2) the LNN precoder outperforms the SINR-max precoder in the presence of CSIT errors as well.

V. Conclusions

We proposed two learning-based twisted convolution precoders for Zak-OTFS that improved channel localization beyond the SINR-max precoder reported in the recent literature. We also introduced a lightweight CNN-based receiver that jointly performs channel estimation and symbol detection, suppressing residual leakage. Extending the precoder framework to multiuser scenarios is a promising future research direction. Precoder designs using superimposed spread pilots and joint precoding across multiple users accounting for the inter-user interference can be explored.

References

[1] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS – a mathematical foundation for communication and radar sensing in the delay-Doppler domain," *IEEE BITS the Inform. Theory Mag.*, vol. 2, no. 2, pp. 36-55, 1 Nov. 2022.

[2] S. K. Mohammed, R. Hadani, A. Chockalingam, and R. Calderbank, "OTFS – predictability in the delay-Doppler domain and its value to communication and radar sensing," *IEEE BITS the Inform. Theory Mag.*, vol. 3, no. 2, pp. 7-31, Jun. 2023.

[3] S. K. Mohammed, et al., "Precoded Zak-OTFS for per-carrier equalization," arXiv:2507.09894v2 [eess.SP] 18 Jul 2025.

[4] R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," *AAAI Conf. Artif. Intell.*, vol. 35, no. 9, pp. 7657-7666, May 2021.

[5] M. A. Girnyk, "Deep-learning based linear precoding for MIMO channels with finite-alphabet signaling," *Phys. Commun.*, vol. 48, p. 101402, 2021.

[6] 3GPP TR 38.901, "Study on channel model for frequencies from 0.5 to 100 GHz," version 16.1.0, Release 16.

[7] D. Mi, M. Dianati, L. Zhang, S. Muhaidat, R. Tafazolli, "Massive MIMO performance with imperfect channel reciprocity and channel estimation error," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3734–3749, Sep. 2017.

[5]CSIT is obtained from the uplink pilot response using the simple model-free read-off procedure in [2]. Assuming channel reciprocity, the slowly varying DD-domain channel allows this uplink estimate to be used for downlink precoding over the DD-channel coherence interval.

[6]With Tx-Rx mismatch, the downlink I/O relation can be expressed as $y_{\text{dd}}^{w_{\text{rx}},\text{dl}}(\tau,\nu) = h_{\text{a}}^{\text{dl}}(\tau,\nu) *_\sigma s_{\text{dd}}(\tau,\nu) + n_{\text{dd}}^{\text{dl}}(\tau,\nu)$, where $h_{\text{a}}^{\text{dl}}(\tau,\nu) = \alpha\big(h_{\text{eff}}^{\text{ul}}(\tau,\nu) *_\sigma a^{\text{ul}}(\tau,\nu)\big)$, where $\alpha = g_t/g_r$ captures the RF reciprocity mismatch, with $g_x = A_x e^{j\phi_x}$, $x \in \{t, r\}$, denoting the Tx and Rx RF chain responses. The uplink effective channel is given by $h_{\text{eff}}^{\text{ul}}(\tau,\nu) = g_r h_{\text{eff}}(\tau,\nu)$, based on which the precoder $a^{\text{ul}}(\tau,\nu)$ is designed. Following [7], A_x and ϕ_x are modeled as truncated Gaussian random variables. For Fig. 3(a), we consider $\sigma_A^2 = \sigma_\phi^2 = 2$, with amplitude and phase truncation intervals of $[-12, 12]$ dB and $[-20°, 20°]$, respectively.